

stack during dynamic mapping. In the case of static mapping, a stack map is generated and stored as the method is compiled on the heap. A typical compiled method shape for a Java method is illustrated schematically in Figure 4. The compiled method is made up of a number of fields, each four bytes in length, including the object header 400, bytecodes 402, start PC 404, class pointers 406, selector 408, Java flags 410 and literals 414. According to the invention, the compiled method also includes a field for the stack map 412. The stack map field 412 includes an array that encodes the information about the temps or local variables in the method generated by the stack mapper in the manner described above, and a linear stack map list that a garbage collector can use to access the stack shape for a given destination PC in the array by calculating the offset and locating the mapping bits in memory.

IN THE ABSTRACT:

Please amend the Abstract of the Invention as follows:

AS --The stack mapper of the present invention seeks to determine the shape of the stack at a given program counter. This is accomplished by locating all start points possible for a given method, that is, at all of the entry points for the method and all of the exception entry points, and trying to find a path from the beginning of the method to the program counter in question. The mapper first tries to locate a linear path from the beginning of the method, and then iteratively processes the sequence of bytes at each branch until the destination program counter is reached. Once the path is found, a simulation is run of the stack through that path, which is used as the virtual stack for the purposes of the garbage collector.
